#2 Priority 6/12/01

Docket No. 1448.1012/HJS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Kiyosi ITO et al.

Group Art Unit:

Serial No.:

Examiner:

Filed: March 13, 2001

For: COMPILER PARALLELIZING SCHEDULE METHOD

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application(s):

Japanese Patent Application No. 2000-309759
Filed: October 10, 2000

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date, as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 13, 2001

By: _____

H. J. Staas
Registration No. 22,010

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

日　本　国　特　許　庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年１０月１０日

出願番号
Application Number: 特願２０００－３０９７５９

出願人
Applicant (s): 富士通株式会社

２００１年　２月　９日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号　出証特２００１－３００５７６３

【書類名】　特許願

【整理番号】　0000912

【提出日】　平成12年10月10日

【あて先】　特許庁長官殿

【国際特許分類】　G06F 15/16

G06F 9/45

【発明の名称】　コンパイラ並列化スケジュール方法

【請求項の数】　9

【発明者】

【住所又は居所】　東京都品川区大崎２丁目８番８号　富士通デバイス株式会社内

【氏名】　伊藤　清志

【発明者】

【住所又は居所】　東京都品川区大崎２丁目８番８号　富士通デバイス株式会社内

【氏名】　冨居　義仁

【発明者】

【住所又は居所】　東京都品川区大崎２丁目８番８号　富士通デバイス株式会社内

【氏名】　岩間　芳樹

【発明者】

【住所又は居所】　東京都品川区大崎２丁目８番８号　富士通デバイス株式会社内

【氏名】　植草　直幸

【発明者】

【住所又は居所】　東京都品川区大崎２丁目８番８号　富士通デバイス株式会社内

【氏名】　佐藤　恭央

【特許出願人】

【識別番号】　000005223

【氏名又は名称】　富士通株式会社

【代理人】

【識別番号】　100104190

【弁理士】

【氏名又は名称】　酒井　昭徳

【手数料の表示】

【予納台帳番号】　041759

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9906241

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　コンパイラ並列化スケジュール方法

【特許請求の範囲】

【請求項１】　命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【請求項２】　命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

同じプライオリティ値の命令間に発行制限による遅延があるか否かを調べる工程と、

発行制限による遅延がある場合に、前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

前記新プライオリティ値に基づいて命令の発行順序を決定し、前記各命令のスロット配置をおこなう工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【請求項３】　発行制限による遅延がある命令群を最適化対象グループとし、当該最適化対象グループに含まれる複数の命令に共通の先行命令をネック命令とし、当該ネック命令と前記最適化対象グループとの間で前記逆プライオリティ値を求めることを特徴とする請求項２に記載のコンパイラ並列化スケジュール方法。

【請求項４】　前記重み付けは、前記最適化対象グループから前記ネック命

令までの命令に付与される第１の重み値と、前記ネック命令よりも先の先行命令に付与される第２の重み値とからなることを特徴とする請求項３に記載のコンパイラ並列化スケジュール方法。

【請求項５】　前記最適化対象グループに含まれる複数の命令に対して、前記逆プライオリティ値の小さい順、前記逆プライオリティ値が同じ場合には前記先行命令数の少ない順、前記逆プライオリティ値と前記先行命令数が同じ場合には行番号の早い順、または前記逆プライオリティ値と前記先行命令数と前記行番号が同じ場合には生成順番の早い順に、優先順位を設定し、その優先順位にしたがって、前記第１の重み値を決定することを特徴とする請求項４に記載のコンパイラ並列化スケジュール方法。

【請求項６】　前記優先順位にしたがって、第１番目の命令の第１の重み値を、前記最適化対象グループ内の命令を実際の発行制限を考慮して発行するのに要する命令数から１を引いた値とし、第２番目以降の命令の第１の重み値を前記命令数から１を引いた値から順次１ずつ小さくした値とすることを特徴とする請求項５に記載のコンパイラ並列化スケジュール方法。

【請求項７】　前記最適化対象グループ内の各命令に対する先行命令の第１の重み値を、当該先行命令につづく後続命令の第１の重み値をそのまま継承した値とし、前記後続命令が複数存在する場合には最も大きい第１の重み値を継承した値とすることを特徴とする請求項５または６に記載のコンパイラ並列化スケジュール方法。

【請求項８】　前記ネック命令の先行命令の第２の重み値を、前記ネック命令に対応する最適化対象グループ内の命令を実際の発行制限を考慮して発行するのに要する命令数に等しい値とすることを特徴とする請求項４に記載のコンパイラ並列化スケジュール方法。

【請求項９】　前記ネック命令の先行命令の第２の重み値を、前記ネック命令に対応する最適化対象グループとは別の最適化対象グループに起因して新たな第２の重み値が発生した場合には、当該第２の重み値を加算した値とすることを特徴とする請求項４に記載のコンパイラ並列化スケジュール方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、コンパイラ並列化スケジュール方法に関し、特にＶＬＩＷ（Ｖｅｒｙ　Ｌｏｎｇ　Ｉｎｓｔｒｕｃｔｉｏｎ　Ｗｏｒｄ）アーキテクチャ用コンパイラの並列化スケジュール方法に関する。

【０００２】

従来より、並列に動作し得る複数の演算ユニットを持つ計算機（コンピュータ）用の目的コードを生成するコンパイラにおいて、目的コードの効率を高めるため、複数の演算ユニットにおいて異なる命令を並列に処理するようなスケジューリングがおこなわれている。

【０００３】

【従来の技術】

従来、コンパイラ並列化スケジュール方法として、特開平１０－２０７８５４号公開公報に記載されたものがある。このスケジュール方法は、命令間の依存関係と並列動作抑制関係を依存・並列動作抑制グラフで表し、それに基づいてパスレイテンシおよび並列動作抑制数を計算し、その計算結果が追加された依存・並列動作抑制グラフに基づいて目的コードの並列なスケジューリングをおこなうものである。

【０００４】

【発明が解決しようとする課題】

しかしながら、上述した従来のスケジュール方法では、全命令のノードに対して並列動作抑制関係と並列動作抑制数を調べるため、その処理に時間がかかってしまい、目的コードの並列化処理に要する時間が長くなるという欠点がある。また、上述した従来のスケジュール方法では、命令に付与する重み値が並列動作抑制数の１種類であるため、並列度が十分に高いとはいえず、さらなる並列度の向上を図る余地がある。

【０００５】

本発明は、上記問題点に鑑みてなされたものであって、並列に動作し得る複数の演算ユニットを持つ計算機用の目的コードを生成するコンパイラにおいて、目

的コードの並列度を向上させるとともに、より高速に目的コードの並列化処理をおこなうことができるコンパイラ並列化スケジュール方法を提供することを目的とする。

【０００６】

【課題を解決するための手段】

上記目的を達成するため、本発明にかかるコンパイラ並列化スケジュール方法は、従来のプライオリティ値ごとに命令を分類し、プライオリティ値が同じ命令間で発行制限があるか否かを調べ、さらに発行制限がある命令群について発行制限による遅延があるか否かを調べ、発行制限による遅延がある命令群よりなる最適化対象グループから、その最適化対象グループ内の命令に共通の先行命令であるネック命令までの命令に対して逆プライオリティ計算をおこない、その逆プライオリティ値に基づいて、最適化対象グループからネック命令までの命令に第１の重み値（アドバンテージ）を付与し、一方、ネック命令よりも先の先行命令に第２の重み値（ウェイト）の重み値を付与し、それら重み値を考慮して再度プライオリティ計算をおこない、その新プライオリティに基づいて各命令に対してスロット配置をおこなうことを特徴とする。

【０００７】

この発明によれば、従来のプライオリティ値ごとに分類した命令群について、各命令群内の命令間で発行制限があるか否かを調べ、発行制限がある命令群について発行制限による遅延があるか否かを調べるため、全命令のノードに対して並列動作抑制関係を調べるのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【０００８】

また、逆プライオリティ計算および第１の重み値（アドバンテージ）を付与するための計算を、最適化対象グループからネック命令までの範囲でおこなうため、全命令のノードに対して並列動作抑制数を計算するのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【０００９】

さらに、最適化対象グループからネック命令までの命令に第１の重み値（アド

バンテージ）を付与し、ネック命令よりも先の先行命令に第２の重み値（ウェイト）を付与するため、命令に付与する重み値が並列動作抑制数だけである場合に比べて、目的コードの並列度を向上させることができる。

【００１０】

【発明の実施の形態】

以下に、本発明の実施の形態について図１～図１０を参照しつつ詳細に説明する。

【００１１】

（処理の概要の説明）

まず、本発明にかかるコンパイラ並列化スケジュール方法の概要について、単純化したモデルを例にして説明する。図１は、本発明にかかるコンパイラ並列化スケジュール方法の概要を示すフローチャートである。また、図２～図８は、本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図または図表である。

【００１２】

まず、命令間の依存関係を表す有向グラフ（以下、ＤＡＧとする。ＤＡＧ：ｄｉｒｅｃｔｅｄ　ａｃｙｃｌｉｃ　ｇｒａｐｈ）を生成し（ステップＳ１１）、プライオリティ計算をおこなう（ステップＳ１２）。プライオリティ計算をおこなうにあたっては、まず、ステップＳ１１で生成したＤＡＧに基づいて従来のプライオリティ計算をおこなう（ステップＳ１５）。

【００１３】

図２に、説明のため単純化したサンプルソースのモデルと、そのソースについてＤＡＧ生成およびプライオリティ計算をおこなった結果を示す。図２において、中央の矢印を挟んで左側がモデルとなるサンプルソースであり、右側がそのモデルのＤＡＧおよびプライオリティ値を示している。

【００１４】

図２に示すモデルにおいて、命令１（図に①で示す）は命令２（図に②で示す）と依存関係にあり、また命令３（図に③で示す）は命令４（図に④で示す）と依存関係にある。命令５（図に⑤で示す）と他の命令との間には依存関係がない

。ＤＡＧにおいて、命令１と命令２を結ぶ線、および命令３と命令４を結ぶ線はそれぞれ前述した依存関係を表している（他の図のＤＡＧにおいても同じ）。さらに、命令２、命令４および命令５はストア命令であり、たとえば２個のＩ０スロット，Ｉ１スロットのうちＩ０スロットでしか発行することができないという制限があると仮定する。

【００１５】

従来のプライオリティ計算の結果、命令２、命令４および命令５のプライオリティ値は１、命令１および命令３のプライオリティ値は２である。これは、従来のプライオリティ計算では、後続の命令を従えていない命令のプライオリティ値を１とし、後続の命令から先行命令の方向に順にたとえば１ずつ加算していくからである。ここで、加算する値が１であるのは、ある命令のプライオリティ値は、その命令の後続命令のプライオリティ値に発行レイテンシとペナルティを加算して求めるが、ここでは単純化のため発行レイテンシを１、ペナルティを０（ゼロ）と仮定しているからである。なお、ＤＡＧにおいて、各命令のノードを表す①～⑤の右下の数字は各命令のプライオリティ値を表す。

【００１６】

図１に戻り、従来のプライオリティ計算（ステップＳ１５）につづいて、発行制限の最適化処理を実行するか否かの判断をおこなう（ステップＳ１６）。具体的には、図３に示すように、従来のプライオリティ計算で得られたプライオリティ値ごとに命令を分類したプライオリティ表を作成する。そして、同じプライオリティ値の命令を、スロットの配置制限を考慮して配置するのに要するＶＬＩＷ命令数（これを、実ＶＬＩＷ命令数とする）を求める。

【００１７】

また、同じプライオリティ値の命令数と同じ数の命令を、スロットの配置制限がない場合に発行するのに必要なＶＬＩＷ命令数（これを、最小ＶＬＩＷ命令数とする）を求める。ここで、最小ＶＬＩＷ命令数はつぎの（１）式より求められる。

【００１８】

［最小ＶＬＩＷ命令数］＝［同じプライオリティ値の命令数］／［スロット数］

＋［余り］　・・・（１）

【００１９】

実ＶＬＩＷ命令数と最小ＶＬＩＷ命令数とを比較し、同じプライオリティ値の命令間に発行制限があるか否かを調べる。実ＶＬＩＷ命令数と最小ＶＬＩＷ命令数との比較の結果、つぎの（２）式を満たす場合に発行制限の最適化処理が必要であると判断する。（２）式を満たさない場合に発行制限の最適化処理は不要であると判断する。

【００２０】

［実ＶＬＩＷ命令数］＞［最小ＶＬＩＷ命令数］　・・・（２）

【００２１】

図２に示すモデルの場合、図４に示すように、プライオリティ値が２の命令群では、実ＶＬＩＷ命令数および最小ＶＬＩＷ命令数はともに１であり、上記（２）式を満たさないので、発行制限の最適化処理は不要である。しかし、プライオリティ値が１の命令群では、実ＶＬＩＷ命令数は３であり、それに対して最小ＶＬＩＷ命令数は２であるため、上記（２）式が成り立つ。したがって、このモデルの場合、発行制限の最適化処理が必要となる。

【００２２】

図１に戻り、発行制限の最適化処理をおこなう（ステップＳ１７）。具体的には、まず、各命令の最短命令終了時刻を意味する逆プライオリティ値を求める。逆プライオリティ値は、先行命令がない命令のプライオリティ値を逆プライオリティ１とし、その命令から後続命令の方向に順にたとえば１ずつ加算することにより求められる。この計算を逆プライオリティ計算と呼ぶ。ここで、加算する値が１であるのは、ある命令の逆プライオリティ値は、つぎの（３）式より求められるが、ここでは単純化のため現在のＤＡＧのレイテンシを１とし、先行命令と現在のＤＡＧのペナルティを０（ゼロ）と仮定しているからである。

【００２３】

［逆プライオリティ値］＝［先行する命令群の中で最大の逆プライオリティ値］＋［現在のＤＡＧのレイテンシ］＋［先行命令と現在のＤＡＧのペナルティ］　・・・（３）

【００２４】

図２に示すモデルに対する逆プライオリティ計算の結果を図５に示す。図５に示すＤＡＧにおいて、各命令のノードを表す①～⑤の右下の数字は各命令の従来のプライオリティ値であり、左下の数字は各命令の逆プライオリティ値である。図５に示すように、命令１、命令３および命令５の逆プライオリティ値は１であり、命令２および命令４の逆プライオリティ値は２である。

【００２５】

つづいて、発行制限があるプライオリティ値が１の命令群、すなわち命令２、命令４および命令５について重み値を付ける。その際、逆プライオリティ値が最小の命令に対して、実ＶＬＩＷ命令数から１を引いた値を重み値として付与する。そして、逆プライオリティ値が大きくなるにつれて順に重み値を１ずつ小さくする。なお、逆プライオリティ値が同じ命令どうしでは、生成が早いものの方が重み値が大きくなるようにする。

【００２６】

図２に示すモデルについて、実ＶＬＩＷ命令数が３であるため、図６にまとめて示すように、逆プライオリティ値が一番小さい命令５の重み値は２である。また、逆プライオリティ値が２である命令２と命令４のうち生成順が早い命令２の重み値は１である。したがって、命令４の重み値は０（ゼロ)である。

【００２７】

つづいて、図１に戻り、発行制限がある命令群に含まれる各命令に付与された重み値を考慮して、再度プライオリティ計算をおこない、新プライオリティ値を取得する（ステップＳ１８）。ここで、新プライオリティ値はつぎの（４）式より得られる。

【００２８】

［新プライオリティ値］＝［後続命令のプライオリティ値］＋［発行レイテンシ］＋［ペナルティ］＋［重み値］　・・・（４）

【００２９】

なお、上述したように、このモデルでは、単純化するため、発行レイテンシを１とし、ペナルティを０（ゼロ)としている。図２に示すモデルについて、図７

に示すＤＡＧの各命令のノードを表す①～⑤の右下の数字が各命令の新プライオリティ値である。すなわち、命令１および命令５の新プライオリティ値は３である。命令２および命令３の新プライオリティ値は２である。命令４の新プライオリティ値は１である。

【００３０】

図１に戻り、各命令の新プライオリティ値に基づいて、レディリストを生成し（ステップＳ１３）、スロット配置をおこない（ステップＳ１４）、スケジューリングを終了する。図２に示すモデルでは、図８に示すように、命令５、命令１、命令２、命令３、命令４の順にスロット配置がおこなわれる。したがって、コンパイラの内部テーブル（擬似機械テーブル）において、あるサイクルにおいてＩ０スロットおよびＩ１スロットからそれぞれ命令５および命令１が発行され、つぎのサイクルでＩ０スロットおよびＩ１スロットからそれぞれ命令２および命令３が発行され、さらにつぎのサイクルでＩ０スロットから命令４が発行される配置となる。

【００３１】

ところで、ステップＳ１６で発行制限の最適化処理を実行しないと判断した場合には、ステップＳ１５で計算した旧プライオリティ値をそれ以降の処理のプライオリティ値とし（ステップＳ１９）、それを用いてレディリストの生成（ステップＳ１３）およびスロット配置（ステップＳ１４）をおこない、スケジューリングを終了する。

【００３２】

（処理の詳細な説明）

つぎに、本発明にかかるコンパイラ並列化スケジュール方法について、より具体的なモデルを例にして詳細に説明する。図９は、本発明にかかるコンパイラ並列化スケジュール方法を詳細に示すフローチャートである。また、図１０～図１９は、本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図または図表である。

【００３３】

まず、処理の概要において説明したように、ＤＡＧを生成し、従来のプライオ

リティ計算をおこなう（ステップＳ９１）。図１０に、モデルとなるＤＡＧおよび各命令のプライオリティ値の例を示す。このモデルでは、命令１（図に①で示す）は命令２（図に②で示す）、命令４（図に④で示す）および命令６（図に⑥で示す）と依存関係にある。また、命令２は命令３（図に③で示す）と依存関係にある。命令４は命令５（図に⑤で示す）と依存関係にある。さらに、命令３、命令５および命令６はストア命令であり、たとえば２個のＩ０スロット，Ｉ１スロットのうちＩ０スロットでしか発行することができないという制限があるとする。また、単純化するため、発行レイテンシおよびペナルティをそれぞれ１および０（ゼロ)とする。

【００３４】

従来のプライオリティ計算の結果、命令１のプライオリティ値は３、命令２および命令４のプライオリティ値は２、命令３、命令５および命令６のプライオリティ値は１である。従来のプライオリティ計算については処理の概要において説明したので、省略する。なお、図１０に示すＤＡＧにおいて、各命令のノードを表す①～⑥の右下の数字は各命令のプライオリティ値を表す。つづいて、図９に戻り、従来のプライオリティ値ごとに命令を分類してプライオリティ表を作成する（ステップＳ９２）。作成したプライオリティ表を図１１に示す。

【００３５】

図９に戻り、プライオリティ値が１、２および３の各グループについて、そのグループに属する命令間で発行制限があるか否かを調べる（ステップＳ９３）。図１０に示すモデルでは、図１２に示すように、プライオリティ１のグループに発行制限があり、プライオリティ２およびプライオリティ３の各グループには発行制限がない。したがって、図９に戻り、プライオリティ１のグループについて、発行制限による遅延があるか否かを調べる（ステップＳ９４）。なお、プライオリティ２およびプライオリティ３の各グループについては、発行制限がないため、発行制限による遅延の有無を調べる必要はない。

【００３６】

発行制限による遅延の有無を調べるため、実ＶＬＩＷ命令数と最小ＶＬＩＷ命令数を求め、それらを比較する。図１０に示すモデルでは、図１３および図１４

に示すように、命令３、命令５および命令６はＩ０スロットでしか発行することができないため、実ＶＬＩＷ命令数は３となる。それに対して、最小ＶＬＩＷ命令数は２であるので、前記（２）式を満たす。したがって、プライオリティ１のグループでは発行制限による遅延が発生することとなり、プライオリティ１のグループは発行制限の最適化処理をおこなう対象となる。本明細書では、発行制限の最適化処理をおこなう対象となるプライオリティのグループを最適化対象グループと呼ぶことにする。ここまでの結果を図１５にまとめて示す。

【００３７】

つづいて、図９に戻り、ネック命令を取得する（ステップＳ９５）。ここで、ネック命令とは、最適化対象グループ（図１０に示すモデルではプライオリティ１のグループ）に含まれる命令の共通の先行命令のことであり、後述する重み付けをおこなう際の有効範囲を求めるために用いられる。最適化対象グループ内の命令は、ネック命令が発行された時点で互いに独立した関係、すなわち相互に依存しない関係となる。

【００３８】

なお、ネック命令は最適化対象グループ内の全ての命令について共通である必要はなく、最適化対象グループ内の一部の命令群について共通の先行命令であってもよい。ある最適化対象グループに共通の先行命令が複数存在する場合には、それら複数の共通の先行命令のうち最小のプライオリティ値を有する命令をネック命令とし、そのネック命令のプライオリティ値をネックプライオリティ値と呼ぶ。図１０に示すモデルでは、最適化対象グループはプライオリティ１のグループであり、また図１６に示すように、ネック命令は命令１であり、ネックプライオリティ値は３である。

【００３９】

つづいて、図９に戻り、前記（３）式に基づいて、逆プライオリティ計算をおこなう（ステップＳ９６）。なお、図１０に示すモデルのように、ネック命令が存在する場合には、ネック命令の逆プライオリティ値は１となる。つまり、図１０に示すモデルではネック命令よりも先の先行命令は存在しないが、ネック命令よりも先に先行命令が存在する場合であっても、ネック命令よりも先の先行命令

については逆プライオリティ計算をおこなわない。

【００４０】

図１７に示すように、命令１はネック命令であるため、その逆プライオリティ値は１となる。命令２、命令４および命令６の逆プライオリティ値は２であり、命令３および命令５の逆プライオリティ値は３である。図１７に示すＤＡＧにおいて、各命令のノードを表す①～⑥の右下の数字は逆プライオリティ値を表す。

【００４１】

つづいて、図９に戻り、最適化対象グループに含まれる複数の命令を逆プライオリティ値の小さい順に並べ替える（ステップＳ９７）。その際、逆プライオリティ値が同じ場合には先行命令数の少ない順とし、逆プライオリティ値および先行命令数が同じ場合には行番号の早い順とし、さらに逆プライオリティ値、先行命令数および行番号が同じ場合には生成順番の早い順とする。したがって、図１０に示すモデルでは、命令６、命令３、命令５の順となる（図１７参照）。

【００４２】

つづいて、各命令に対して重み付けをおこなう（ステップＳ９８）。重み付けには第１の重み値アドバンテージと第２の重み値ウェイトの２種類がある。アドバンテージは、最適化対象グループからネック命令までの命令に付与される。ウェイトは、ネック命令よりも先の先行命令に付与される。なお、図１０に示すモデルでは、ネック命令である命令１よりも先には命令がないため、ウェイトは発生しない。したがって、ここではアドバンテージについてのみ説明し、ウェイトについての説明は後述する。

【００４３】

最適化対象グループ内に含まれる各命令に対し、実ＶＬＩＷ命令数から１を引いた値をアドバンテージの最初の値とし、ステップＳ９７で並べ替えた順に一命令ごとにアドバンテージの値を１ずつ小さくする。最適化対象グループ内に含まれる各命令の先行命令のアドバンテージについては、その先行命令につづく後続命令のアドバンテージをそのまま継承する。後続命令が複数存在する場合には、最も大きいアドバンテージを継承する。

【００４４】

図１０に示すモデルでは、上述したように実ＶＬＩＷ命令数が３であり、かつステップＳ９７で命令６、命令３、命令５の順で並べ替えをおこなったため、図１８に示すように、命令６、命令３および命令５のアドバンテージは２、１および０（ゼロ）となる。そして、命令２のアドバンテージは命令３を継承するため１である。同様に、命令４のアドバンテージは命令５を継承するため０（ゼロ）である。命令１のアドバンテージは、命令２と命令４と命令６のうち最も大きいアドバンテージである命令６を継承するため２である。

【００４５】

つづいて、図９に戻り、ステップＳ９８でおこなった重み付けを考慮して、再度プライオリティ計算をおこなう（ステップＳ９９）。ここでのプライオリティ計算は前記（４）式にしたがうが、後続命令のプライオリティ値に発行レイテンシとペナルティを加算したものは、ステップＳ９１で求めた従来のプライオリティ値と同じである。つまり、新プライオリティ値は、今までのプライオリティ値に重み値を加算したものである。したがって、図１９の上段に示すように、命令１の新プライオリティ値は５（＝３＋２）、命令２の新プライオリティ値は３（＝２＋１）、命令３の新プライオリティ値は２（＝１＋１）、命令４の新プライオリティ値は２（＝２＋０）、命令５の新プライオリティ値は１（＝１＋０）、命令６の新プライオリティ値は３（＝１＋２）である。

【００４６】

図９に戻り、各命令の新プライオリティに基づいて、レディリストを生成し、スロット配置をおこない（ステップＳ１００）、スケジューリングを終了する。各命令のスロット配置順は、新プライオリティ値が大きい順とし、新プライオリティ値が同じ場合には後続命令の数が少ない順とする。図１０に示すモデルでは、命令１、命令６、命令２、命令３、命令４、命令５の順にスロット配置がおこなわれる。

【００４７】

したがって、図１９の下段右側に示すように、擬似機械テーブルにおいて、あるサイクルにおいてＩ０スロットから命令１が発行される。このときＩ１スロットからは命令が発行されない。そして、つぎのサイクルでＩ０スロットおよびＩ

１スロットからそれぞれ命令６および命令２が発行され、そのつぎのサイクルでＩ０スロットおよびＩ１スロットからそれぞれ命令３および命令４が発行され、さらにつぎのサイクルでＩ０スロットから命令５が発行される配置となる。つまり、命令１～命令６の６個の命令を４サイクルで発行することができる。

【００４８】

ここで、比較のため、図１９の下段左側に、重み付けをおこなう前の従来のプライオリティに基づいてスロット配置をおこなった結果を示す。この場合、命令１、命令２、命令４、命令３、命令５、命令６の順にスロット配置がおこなわれることとなり、全部で５サイクル必要となる。

【００４９】

ところで、ステップＳ９３で発行制限が発生しない場合、またはステップＳ９４で発行制限による遅延がない場合には、ステップＳ９１で計算した従来のプライオリティを用いてレディリストの生成およびスロット配置（ステップＳ１００）をおこない、スケジューリングを終了する。

【００５０】

つぎに、図２０に示すモデルを用いてウェイトについて説明する。図２０に示すモデルでは、命令１（図に①で示す）は命令２（図に②で示す）と依存関係にある。命令２は命令３（図に③で示す）と依存関係にある。命令３は命令４（図に④で示す）と依存関係にある。命令４は命令５（図に⑤で示す）と依存関係にある。命令６（図に⑥で示す）は命令７（図に⑦で示す）と依存関係にある。命令７は命令８（図に⑧で示す）と依存関係にある。命令８は命令４、命令９（図に⑨で示す）および命令１１（図に○11示す）と依存関係にある。命令９は命令１０（図に○10示す）と依存関係にある。

【００５１】

図２０に示すモデルでは、最適化対象グループに含まれる命令は、命令５と命令１０と命令１１であり、ネック命令は命令８である。命令３および命令８の逆プライオリティ値は１である。命令４、命令９および命令１１の逆プライオリティは２である。命令５および命令１０の逆プライオリティ値は３である。

【００５２】

ウェイトは、ネック命令よりも先の先行命令、すなわちネックプライオリティよりも大きなプライオリティ値を有する命令に対して付与される。ウェイトは、そのウェイトが付与される命令のすぐ後の最適化対象グループの実ＶＬＩＷ命令数に等しい。そして、先行命令のウェイトは、後続命令のウェイトをそのまま継承する。したがって、図２０に示すモデルでは、最適化対象グループ（命令５、命令１０および命令１１）の実ＶＬＩＷ命令数は３であるため、図２１に示すように、ネック命令（命令８）よりも先にある命令２および命令７のウェイトは３となる。命令２および命令７よりも先にある命令１および命令６については、命令２および命令７のウェイトを継承するため、ウェイトは３となる。

【００５３】

ここで、ウェイトは、別の最適化対象グループに起因して新たなウェイトが付与されると、それを加算していく。これについて、たとえば図２２に示すモデルを用いて具体的に説明する。図２２に示すモデルでは、命令１（図に①で示す）は命令２（図に②で示す）と依存関係にある。命令２は命令３（図に③で示す）と依存関係にある。命令３は命令４（図に④で示す）と依存関係にある。命令４は命令５（図に⑤で示す）と依存関係にある。命令５は命令６（図に⑥で示す）と依存関係にある。命令６は命令７（図に⑦で示す）と依存関係にある。

【００５４】

命令７は命令８（図に⑧で示す）と依存関係にある。命令９（図に⑨で示す）は命令１０（図に○10で示す）と依存関係にある。命令１０は命令３および命令１１（図に○11で示す）と依存関係にある。命令１１は命令１２（図に○12で示す）と依存関係にある。命令１２は命令１３（図に○13で示す）と依存関係にある。命令１３は命令１４（図に○14で示す）と依存関係にある。命令１４は命令７および命令１５（図に○15で示す）と依存関係にある。命令１５は命令１６（図に○16で示す）と依存関係にある。

【００５５】

このモデルでは、第１の最適化対象グループに含まれる命令は、命令８と命令１６であり、この第１の最適化対象グループに対するネック命令は命令１４である。また、このモデルでは、命令４と命令１２は、命令１０をネック命令とする

第２の最適化対象グループに含まれる。

【００５６】

図２２に示すモデルでは、命令８および命令１６の従来のプライオリティ値は１である。そして、従来のプライオリティ値は、命令８から命令１に向かって、また命令１６から命令９に向かって順に１ずつ大きくなり、命令１および命令９のプライオリティ値は８である。このようなモデルに対して、逆プライオリティ値は、図２３に示すように、命令８と命令１６と命令４と命令１２では３であり、命令７と命令１５と命令３と命令１１では２であり、命令６と命令１４と命令２と命令１０では１である。

【００５７】

図２２に示すモデルに対して付与されるウェイトは、図２４に示すように、命令６～命令８および命令１４～命令１６では０（ゼロ）である。第１の最適化対象グループ内の命令８および命令１６の実ＶＬＩＷ命令数は２であるため、ネック命令である命令１４よりも先の先行命令である命令５および命令１３のウェイトは２となる。このウェイトの値はさらに先の先行命令に継承されるため、命令２～命令４および命令１０～命令１２のウェイトも２となる。さらには、第１の最適化対象グループに起因して付与された２のウェイトはさらに先の命令１および命令９にも継承される。

【００５８】

また、第２の最適化対象グループ内の命令４および命令１２の実ＶＬＩＷ命令数は２であるため、この第２の最適化対象グループに対応するネック命令である命令１０の先行命令である命令１および命令９には、新たに第２の最適化対象グループに起因する２のウェイトが付与される。したがって、命令１および命令９のウェイトは、第１の最適化対象グループに起因して付与された２と、第２の最適化対象グループに起因して付与された２とを足して４となる。なお、図２４に示すＤＡＧにおいて、各命令のノードを表す①～○16の右下の式のうち、「＋」を挟んで左側の数字はウェイトを表し、右側の数字はアドバンテージを表す。各命令に対して付与される実際の重み値は、各命令のウェイトとアドバンテージを加算したものである。

【００５９】

つぎに、本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明する。つぎの（１）～（３）のいずれかの条件に該当する場合には、発行制限の最適化性能が低下するおそれがあるため、発行制限の最適化処理を実施しない。（１）最適化対象グループの命令に先行命令が存在しない。（２）ネック命令が存在するが、最適化対象グループのプライオリティとネックプライオリティとの間に他の命令が全く存在しない。（３）ネック命令が存在するが、最適化対象グループのプライオリティとネックプライオリティとの間に、最適化対象グループの先行命令ではない命令が存在する。

【００６０】

たとえば図２５に示すモデルでは、命令１（図に①で示す）は命令２（図に②で示す）と依存関係にある。命令３（図に③で示す）は命令４（図に④で示す）と依存関係にある。命令５（図に⑤で示す）は命令６（図に⑥で示す）と依存関係にある。命令２、命令４および命令６のプライオリティ値は１である。命令１、命令３および命令５のプライオリティ値は２である。そして、最適化対象グループに含まれる命令は、命令１と命令３と命令５である。このモデルは上述した（１）の条件に該当するため、発行制限の最適化処理をおこなわない。

【００６１】

また、図２６に示すモデルでは、命令１（図に①で示す）は命令２（図に②で示す）と依存関係にある。命令３（図に③で示す）は命令４（図に④で示す）および命令５（図に⑤で示す）と依存関係にある。命令２、命令４および命令５のプライオリティ値は１である。命令１および命令３のプライオリティ値は２である。そして、最適化対象グループに含まれる命令は、命令２と命令４と命令５であり、これに対するネック命令は命令３である。このモデルは上述した（２）の条件に該当するため、発行制限の最適化処理をおこなわない。

【００６２】

また、図２７に示すモデルでは、命令１（図に①で示す）は命令２（図に②で示す）と依存関係にある。命令２は命令３（図に③で示す）と依存関係にある。

命令３は命令４（図に④で示す）と依存関係にある。命令５（図に⑤で示す）は命令６（図に⑥で示す）、命令９（図に⑨で示す）および命令１２（図に○12で示す）と依存関係にある。命令６は命令７（図に⑦で示す）と依存関係にある。命令７は命令８（図に⑧で示す）と依存関係にある。

【００６３】

命令９は命令１０（図に○10示す）と依存関係にある。命令１０は命令１１（図に○11示す）と依存関係にある。命令１２は命令１３（図に○13で示す）と依存関係にある。命令４、命令８、命令１１および命令１３のプライオリティ値は１である。命令３、命令７および命令１０のプライオリティ値は２である。命令２、命令６、命令９および命令１２のプライオリティ値は３である。命令１および命令５のプライオリティ値は４である。そして、最適化対象グループに含まれる命令は、命令３と命令７と命令１０であり、これに対するネック命令は命令５である。このモデルは上述した（３）の条件に該当し、最適化対象グループのプライオリティとネックプライオリティとの間に、最適化対象グループ内の命令と依存関係がない命令、図示例では命令１２が存在するため、発行制限の最適化処理をおこなわない。

【００６４】

上述した実施の形態によれば、従来のプライオリティ値ごとに分類した命令群について、各命令群内の命令間で発行制限があるか否かを調べ、発行制限がある命令群について発行制限による遅延があるか否かを調べるため、全命令のノードに対して並列動作抑制関係を調べるのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【００６５】

また、上述した実施の形態によれば、逆プライオリティ計算およびアドバンテージを付与するための計算を、最適化対象グループからネック命令までの範囲でおこなうため、全命令のノードに対して並列動作抑制数を計算するのに比べて、より高速に目的コードの並列化処理をおこなうことができる。

【００６６】

さらに、上述した実施の形態によれば、最適化対象グループからネック命令ま

での命令にアドバンテージを付与し、ネック命令よりも先の先行命令にウェイトを付与するため、命令に付与する重み値が並列動作抑制数だけである場合に比べて、目的コードの並列度を向上させることができる。

【００６７】

（付記１）　命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【００６８】

（付記２）　命令間の依存関係に基づいて、各命令のプライオリティ値を計算する工程と、

同じプライオリティ値の命令間に発行制限による遅延があるか否かを調べる工程と、

発行制限による遅延がある場合に、前記各命令に対して最短命令終了時刻に相当する逆プライオリティ値を計算する工程と、

前記逆プライオリティ値に基づいて前記各命令に重み付けをおこなう工程と、

前記各命令に付与した重み値および前記各命令の前記プライオリティ値に基づいて、前記各命令に対して新プライオリティ値を計算する工程と、

前記新プライオリティ値に基づいて命令の発行順序を決定し、前記各命令のスロット配置をおこなう工程と、

を含んだことを特徴とするコンパイラ並列化スケジュール方法。

【００６９】

（付記３）　発行制限による遅延がある命令群を最適化対象グループとし、当該最適化対象グループに含まれる複数の命令に共通の先行命令をネック命令とし、当該ネック命令と前記最適化対象グループとの間で前記逆プライオリティ値

を求めることを特徴とする付記２に記載のコンパイラ並列化スケジュール方法。

【００７０】

（付記４）　前記重み付けは、前記最適化対象グループから前記ネック命令までの命令に付与される第１の重み値と、前記ネック命令よりも先の先行命令に付与される第２の重み値とからなることを特徴とする付記３に記載のコンパイラ並列化スケジュール方法。

【００７１】

（付記５）　前記最適化対象グループに含まれる複数の命令に対して、前記逆プライオリティ値の小さい順、前記逆プライオリティ値が同じ場合には前記先行命令数の少ない順、前記逆プライオリティ値と前記先行命令数が同じ場合には行番号の早い順、または前記逆プライオリティ値と前記先行命令数と前記行番号が同じ場合には生成順番の早い順に、優先順位を設定し、その優先順位にしたがって、前記第１の重み値を決定することを特徴とする付記４に記載のコンパイラ並列化スケジュール方法。

【００７２】

（付記６）　前記優先順位にしたがって、第１番目の命令の第１の重み値を、前記最適化対象グループ内の命令を実際の発行制限を考慮して発行するのに要する命令数から１を引いた値とし、第２番目以降の命令の第１の重み値を前記命令数から１を引いた値から順次１ずつ小さくした値とすることを特徴とする付記５に記載のコンパイラ並列化スケジュール方法。

【００７３】

（付記７）　前記最適化対象グループ内の各命令に対する先行命令の第１の重み値を、当該先行命令につづく後続命令の第１の重み値をそのまま継承した値とし、前記後続命令が複数存在する場合には最も大きい第１の重み値を継承した値とすることを特徴とする付記５または６に記載のコンパイラ並列化スケジュール方法。

【００７４】

（付記８）　前記ネック命令の先行命令の第２の重み値を、前記ネック命令に対応する最適化対象グループ内の命令を実際の発行制限を考慮して発行するの

に要する命令数に等しい値とすることを特徴とする付記４に記載のコンパイラ並列化スケジュール方法。

【００７５】

（付記９）　前記ネック命令の先行命令の第２の重み値を、前記ネック命令に対応する最適化対象グループとは別の最適化対象グループに起因して新たな第２の重み値が発生した場合には、当該第２の重み値を加算した値とすることを特徴とする付記４に記載のコンパイラ並列化スケジュール方法。

【００７６】

（付記１０）　同じプライオリティ値の命令間に発行制限がある場合、同じプライオリティ値の命令を実際の発行制限にしたがって発行するのに要する命令数と、同じプライオリティ値の命令を発行制限がないと仮定して発行するのに要する命令数とを求め、前記両命令数を比較して、実際の発行制限にしたがって発行するのに要する命令数の方が大きい場合に発行制限による遅延があると判断することを特徴とする付記２～９のいずれか一つに記載のコンパイラ並列化スケジュール方法。

【００７７】

（付記１１）　前記最適化対象グループの命令に先行命令が存在しない場合、最適化対象グループのプライオリティ値とネック命令のプライオリティ値との間に他の命令が全く存在しない場合、または最適化対象グループのプライオリティ値とネック命令のプライオリティ値との間に、最適化対象グループの先行命令ではない命令が存在する場合のいずれか一つ以上に該当する場合、前記逆プライオリティ値の計算工程、前記重み付け工程および前記新プライオリティ値の計算工程をおこなわずに、最初に求めたプライオリティ値に基づいて命令の発行順序を決定し、各命令のスロット配置をおこなうことを特徴とする付記３に記載のコンパイラ並列化スケジュール方法。

【００７８】

【発明の効果】

本発明によれば、従来のプライオリティ値ごとに分類した命令群について、各命令群内の命令間で発行制限があるか否かを調べ、発行制限がある命令群につい

て発行制限による遅延があるか否かを調べるとともに、逆プライオリティ計算および第１の重み値を付与するための計算を、最適化対象グループからネック命令までの範囲でおこなうため、より高速に目的コードの並列化処理をおこなうことができる。さらに、本発明によれば、最適化対象グループからネック命令までの命令に第１の重み値を付与し、ネック命令よりも先の先行命令に第２の重み値を付与するため、目的コードの並列度を向上させることができる。

【図面の簡単な説明】

【図１】

本発明にかかるコンパイラ並列化スケジュール方法の概要を示すフローチャートである。

【図２】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図３】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表である。

【図４】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表である。

【図５】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図６】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表である。

【図７】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図８】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図である。

【図９】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に示すフローチャートである。

【図１０】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図１１】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図１２】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図１３】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図１４】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図１５】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図１６】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表である。

【図１７】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図１８】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図１９】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図２０】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図２１】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図２２】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図２３】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図２４】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図である。

【図２５】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図である。

【図２６】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図である。

【図２７】

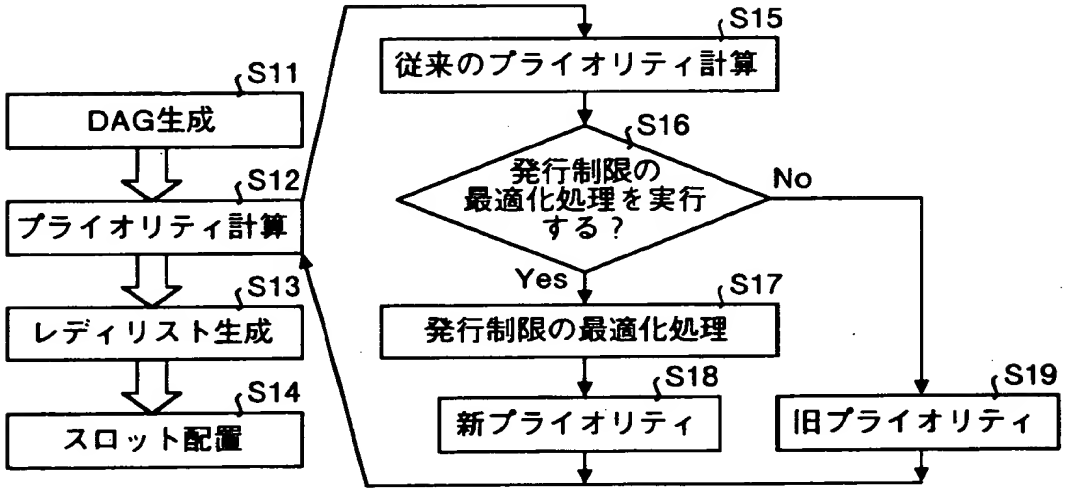
本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化

処理をおこなわない場合について説明するための模式図である。

【書類名】　　図面

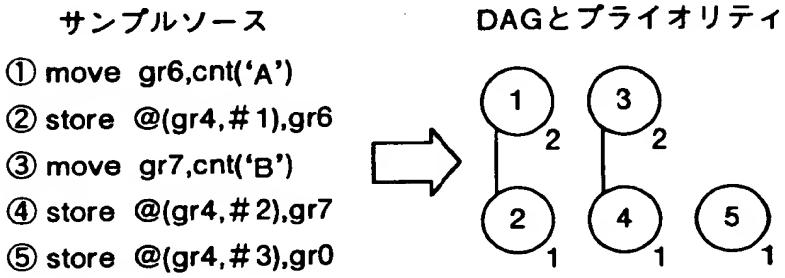
【図１】

本発明にかかるコンパイラ並列化スケジュール方法の概要を示すフローチャート



【図２】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図



【図３】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表

プライオリティ	命令
2	①③
1	②④⑤

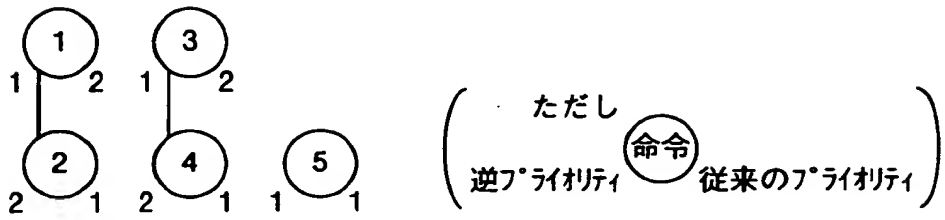
【図４】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表

プライオリティ	実VLIW命令数	最小VLIW命令数	発行制限対応処理必要有無
2	1	1	なし
1	3	2	あり

【図５】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図



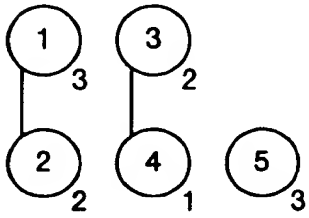
【図６】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための図表

命令	逆ﾌﾟﾗｲｵﾘﾃｨ値	生成順番	重み
②	2	1	1
④	2	2	0
⑤	1	3	2

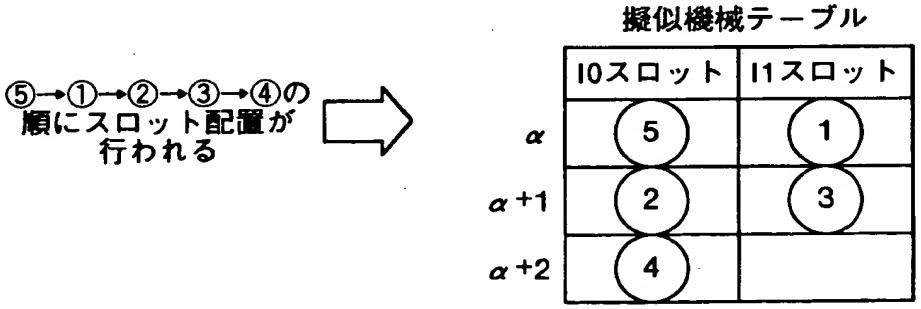
【図７】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図



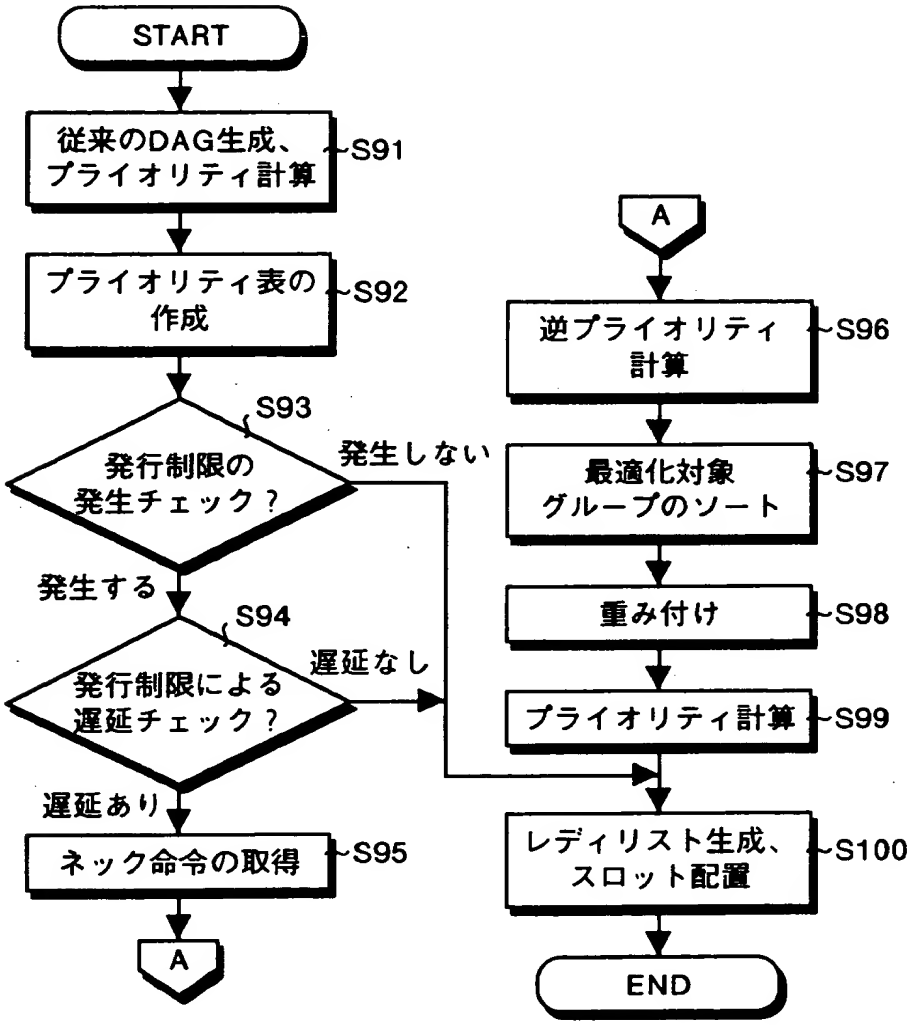
【図８】

本発明にかかるコンパイラ並列化スケジュール方法の概要を説明するための模式図



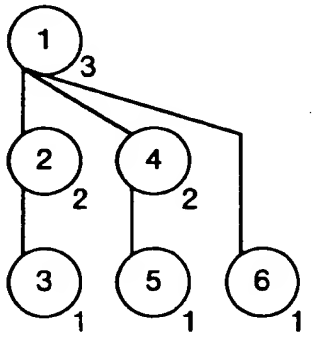
【図９】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に示すフローチャート



【図１０】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



【図１１】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表

プライオリティ	命令リスト
1	③,⑤,⑥
2	②,④
3	①

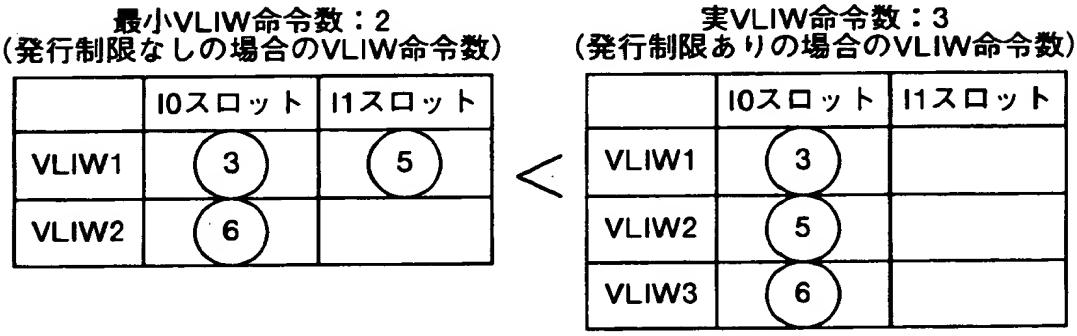
【図１２】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表

プライオリティ	命令リスト	発行制限チェック結果
1	③,⑤,⑥	あり
2	②,④	なし
3	①	なし

【図１３】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



【図１４】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表

発行制限なしの場合のVLIW命令数（最小VLIW命令数）	実際に発行に必要なVLIW命令数（実VLIW命令数）
2個	3個

【図１５】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表

プライオリティ	命令リスト	発行制限チェック結果	遅延チェック
1	③,⑤,⑥	あり	あり
2	②,④	なし	—
3	①	なし	—

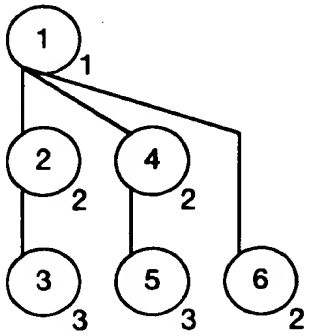
【図１６】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための図表

ネック命令	ネックプライオリティ
①	3

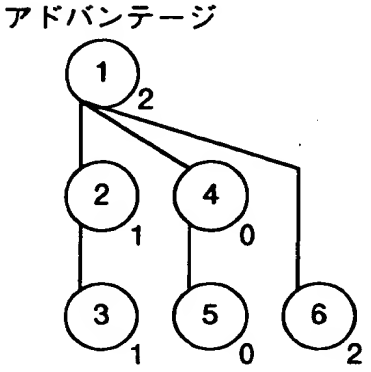
【図１７】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



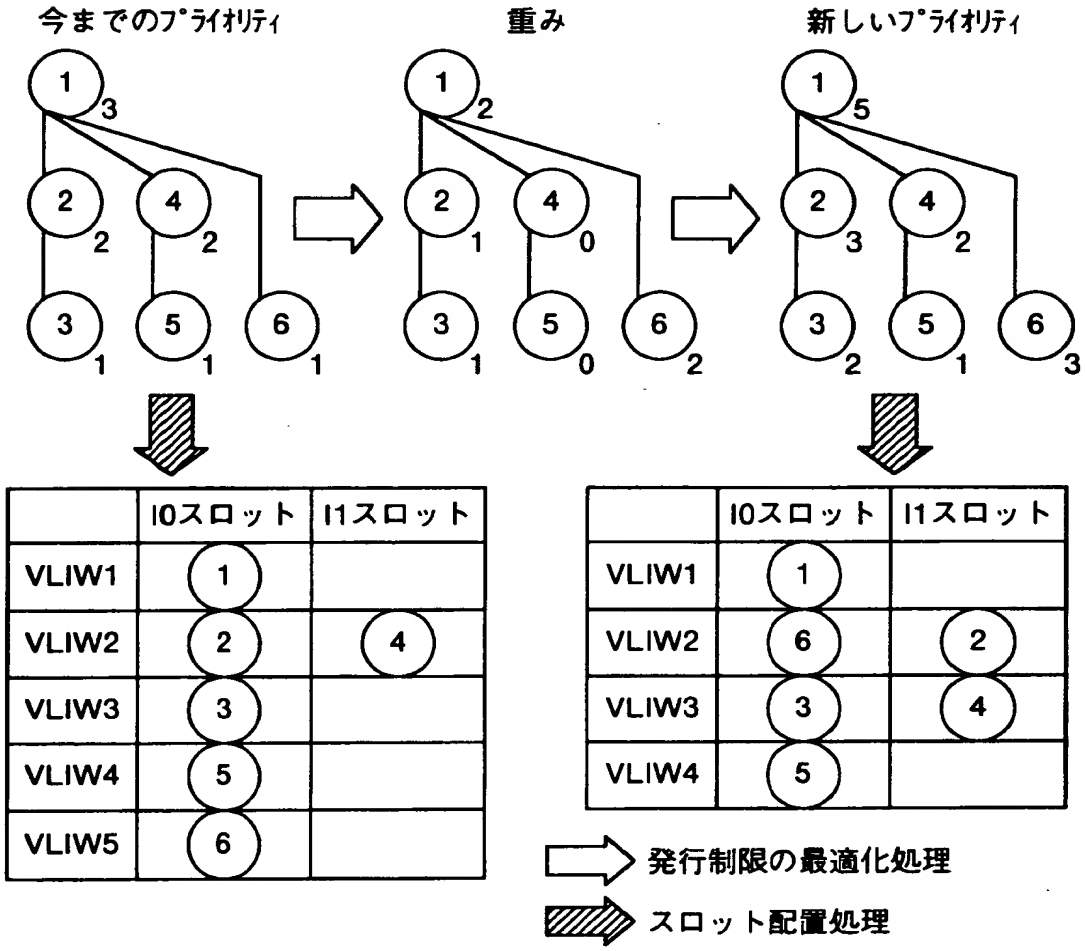
【図１８】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



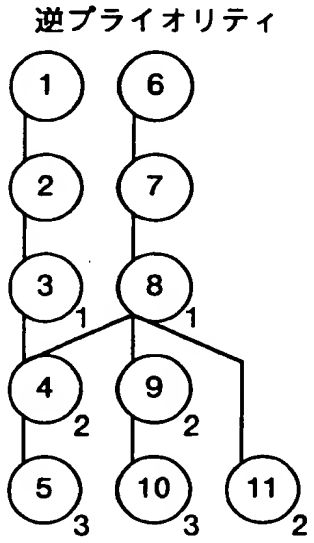
【図１９】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



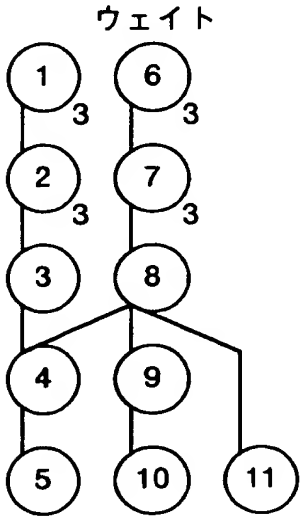
【図２０】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



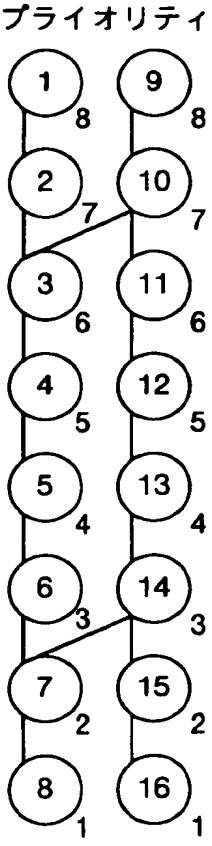
【図２１】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



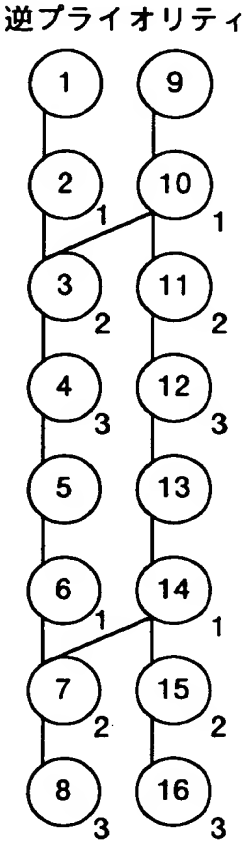
【図２２】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



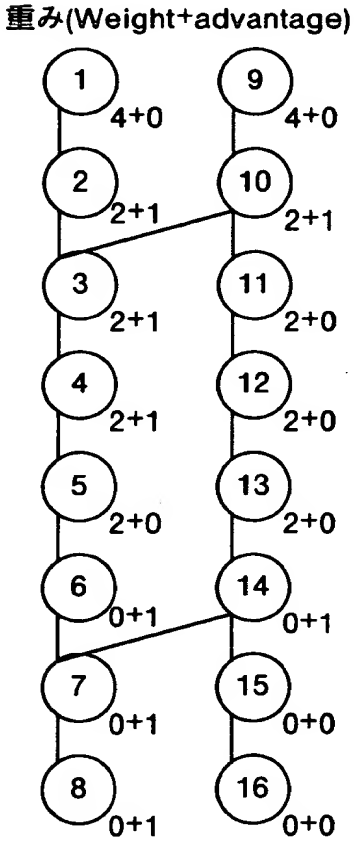
【図２３】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



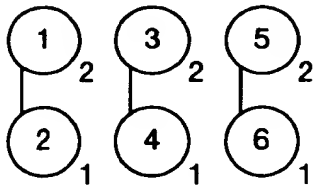
【図２４】

本発明にかかるコンパイラ並列化スケジュール方法を詳細に説明するための模式図



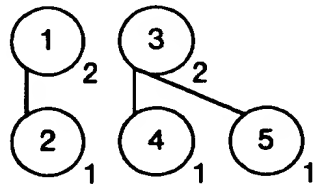
【図２５】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図



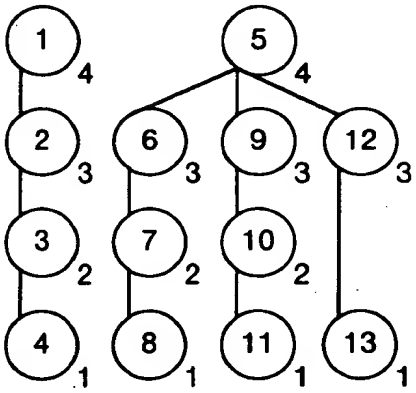
【図２６】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図



【図２７】

本発明にかかるコンパイラ並列化スケジュール方法において発行制限の最適化処理をおこなわない場合について説明するための模式図



【書類名】　　要約書

【要約】

【課題】　並列に動作し得る複数の演算ユニットを持つ計算機用の目的コードを生成するコンパイラ並列化スケジュール方法において、目的コードの並列度を向上させるとともに、より高速に目的コードの並列化処理をおこなうこと。

【解決手段】　従来のプライオリティ値ごとに命令を分類し、プライオリティ値が同じ命令間で発行制限があるかを調べ、発行制限がある命令群について発行制限による遅延があるかを調べ、発行制限による遅延がある命令群よりなる最適化対象グループから、その最適化対象グループ内の命令に共通の先行命令であるネック命令までの命令に対して逆プライオリティ計算をおこない、その逆プライオリティ値に基づき、再度プライオリティ計算をおこない、スロット配置をおこなう。

【選択図】　　図９

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　　富士通株式会社